

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

平1-276189

⑤ Int.Cl.⁴

識別記号

庁内整理番号

⑬ 公開 平成1年(1989)11月6日

G 09 C 1/00

7368-5B

審査請求 未請求 請求項の数 8 (全9頁)

⑭ 発明の名称 暗号方式

⑮ 特 願 昭63-103919

⑯ 出 願 昭63(1988)4月28日

⑰ 発 明 者 宝 木 和 夫 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
⑰ 発 明 者 中 川 聡 夫 茨城県日立市大みか町5丁目2番1号 株式会社日立コントロールシステムズ内
⑰ 発 明 者 佐々木 良一 神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内
⑰ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
⑰ 出 願 人 株式会社日立コントロールシステムズ 茨城県日立市大みか町5丁目2番1号
⑰ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

暗号方式

2. 特許請求の範囲

1. 換字処理とデータ攪乱のための転置処理を組合せて一定長のデータを暗号化するブロック暗号方式において、上記転置処理は、 2^n ビット ($n=2$ 以上の整数) だけ右または左に循環シフトするという操作を組合せることにより行なうことを特徴とする暗号方式。
2. 上記換字処理は、ある定数をXとして、XビットのデータとXビットの鍵データとの間で演算を行なうという操作と、Xビットのデータと該データを右または左に2ビット循環シフトしたものと定数Xとの和を 2^X で割った余りをとるという操作を、組合せることにより行なうことを特徴とする第1項の暗号方式。
3. 上記換字処理と転置処理を組み合わせた暗号処理は、暗号化されたデータと、次の暗号化されるべきデータとの間で演算処理を施した後、

該演算結果をさらに暗号化するというフィードバック処理を有することを特徴とする第1項または第2項の暗号方式。

4. 上記転置処理は、32ビットマイクロコンピュータのソフトウェアを用いて、32ビットのデータを、4ビット、または8ビット、または16ビットだけ右または左に循環シフトするという操作を組合せて行なうことを特徴とする第1項～第3項いずれか1項の暗号方式。
5. 上記転置処理は、16ビットマイクロコンピュータのソフトウェアを用いて、16ビットのデータを、4ビット、または8ビット、右または左に循環するという操作を組合せて行なうことを特徴とする第1項～第3項いずれか1項の暗号方式。
6. 上記転置処理は、8ビットマイクロコンピュータのソフトウェアを用いて、8ビットのデータを、4ビット右または左に循環シフトするという操作により行なうことを特徴とする第1項～第3項いずれか1項の暗号方式。

7. 上記換字処理と転置処理とを組み合わせて、メッセージ認証機能を実現するディジタル署名におけるメッセージ認証コード生成のために必要となるメッセージの圧縮文を生成することを特徴とする第1項～第3項いずれか1項の暗号方式。

8. 換字処理とデータ攪乱のための転置処理を組合せて一定長のデータを暗号化するブロック暗号方式において、上記転置処理は、 2^n ビット ($n=2$ 以上の整数)だけ右または左に循環シフトするという操作を組合せることにより行ない、上記換字処理は、ある定数を X として、 X ビットのデータと X ビットの鍵データとの間で演算を行なうという操作と、 X ビットのデータと該データを右または左に2ビット循環シフトしたものと定数 X との和を 2^X で割った余りをとるという操作を、組合せることにより行ない、メッセージ認証コードを生成するための圧縮暗号に、上記換字処理と転置処理のすくなくとも一つを用いることにより、メッセージ認証コード

の生成と確認をおこなう機能をICカードに組み入れたことを特徴とする暗号方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、コンピュータのメッセージ等を暗号化する装置に関する。

〔従来の技術〕

従来の代表的な暗号アルゴリズムとしては、DES (Data Encryption Standard) と FEAL (Fast Encryption Standard) が知られており、DESに関しては例えば、(1) 小山他、「現代暗号理論」、電子通信学会、pp. 41~49、昭和61年9月において、また、FEALに関しては、(2) 清水他、「高速データ暗号アルゴリズム FEAL」、電子通信学会論文誌D、Vol. J70-D、No. 7、pp. 1413~1423、1987年7月において、それぞれ詳細に述べられている。

先ず、DESの処理における非線形の計算部分、つまりSボックスといわれる処理について説明す

る(上記(1)のp. 45、図3-2とp. 46、図3-3参照)。32ビットのRは、まず、表1に示す拡大型転置表によって置き換えられると共に、一部のビットは重複されて48ビットに拡大されている。

表1 Sボックスの換字表 (S_1)

列 行	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	9	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	6	11	3	14	10	0	6	13

このようにして得られた48ビットのRは、頭から4ビットごとにその後の2ビットを加えた次のような6ビットずつの8組のブロックを形成している。

$R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8$

$R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8$

この48ビットのRは、同じく48ビットの鍵Kと排他的論理和の演算を行ない、6ビットずつ8組に分割して、 S_1 から S_8 までの8つのSボックスに入力する。 $S_1 \sim S_8$ を選択関数と呼ぶ。これらのSボックスは、6ビットを入力して4ビットを出力する。

例として、表2に一つのSボックス S_1 を取り上げてその換字表を示す。

一つのSボックスには、4種類(行番号0, 1, 2, 3)が用意され、この4種類の換字表のどれを用いるかは、入力した6ビットのうち最初と最後のビットを用いて換字表を選ぶ。そして選ばれた換字表にしたがって入力した6ビットの中央の4ビットが換字される。具体的な例として、 S_1 に対して2進数の入力パターンが011011となっている場合、最初の0と最後の1で表わされ

表2 拡大型転置表 E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

ている01、つまり行1(2進数01は10進数1であるから)の換字表が選ばれる。次に中央の4ビットのパターン1101(10進数13)で表わされる列13で指定され、この結果行1、列13で指定される値5、つまり0101が出力されて4ビットの換字パターンとなる。DESではこのような処理 $f(R, K)$ を用い一段の処理を構成し、これを16段繰り返す。

上記の処理例に見られるように、DESは1ビ

不正使用や盗取等に対する情報セキュリティを確保するため、伝送路上のデータやコンピュータに蓄積されたデータを暗号化することは有効な対策であると考えられる。

昭和52年に、米国商務省標準局が暗号アルゴリズムの標準として制定したDESは、データの暗号化を行う一つの手段である。

ところが、DESはビット単位での処理がたいへん多いため、バイト単位の処理を基調とするマイクロコンピュータのソフトで実現しようとする、処理に時間がかかり、実用的な速度が得られなかった。

この問題に対し、上記FEALは、1バイト(8ビット)単位の処理を基調とするため、8ビットマイクロコンピュータで実現する場合、DESに比べ数倍以上の高速化を達成することができた。FEALにより、8ビットマイクロコンピュータのソフトを用いてある程度実用的な速度が得られるようになったと考えられる。

しかし、最近のマイクロエレクトロニクスの技

術単位の処理が基本になっている。

次にFEALの処理における非線形の計算部分、つまり、関数Sを含んでいる部分について説明する(上記(2)のp.1416、図4及び図5参照)。FEALの非線形部はDESの非線形部に比べ、数学的な記述が簡単である。32ビットデータ α は8ビットのデータ $\alpha^0, \alpha^1, \alpha^2, \alpha^3$ にそれぞれ分割された後、8ビットを単位として、鍵データと排他的論理和がとられる。その後、所定の関数Sによる処理が実行される。

$$\text{関数 } S : S(x_1 + x_2 + \delta) = \text{Rot}_2(w)$$

$$\text{ただし、 } w = (x_1 + x_2 + \delta) \bmod 256$$

$$\delta = 0 \text{ または } 1 \text{ (定数)}$$

この処理 $f(\alpha, \beta)$ を用い、一段の処理を構成し、これが8段繰り返される。上記の処理に見られるように、FEALは8ビット単位の処理が基本になっている。

[発明が解決しようとする問題点]

情報処理と通信技術の進歩によるコンピュータ・ネットワークの普及化、大衆化に伴い、データの

術の進歩によって、8ビットマイクロコンピュータよりも16ビットマイクロコンピュータ、さらに、16ビットマイクロコンピュータよりも32ビットマイクロコンピュータが使われ出している。近い将来、32ビットマイクロコンピュータが使われる割合がたいへん大きくなると予想されている。32ビットマイクロコンピュータの時代になると、さらに高速の暗号処理が要求されるものと予想される。ところが、32ビットマイクロコンピュータは4バイト基調の処理を行うため、1バイト基調の8ビットマイクロコンピュータ用に設計されたFEALを32ビットマイクロコンピュータで実現しようとする、非効率であった。

そこで、32ビットマイクロコンピュータ向けの4バイト基調の処理を行う暗号アルゴリズムが望まれていた。

[問題点を解決するための手段]

上記の問題点を解決するため、次の手段を用いる。

すなわち、32ビットマイクロコンピュータと

メモリからなる暗号変換装置において、

暗号変換されるべきデータの換字変換処理を、
32ビットのデータxとy同士の演算、

$$x + y,$$

つまり、xとyを加算し、 2^{32} で割った余りをと
るという処理と、

$$\text{Rot}_2(x) + x + \alpha,$$

つまり、32ビットのデータxを左または右に2
ビット循環シフトした後、その結果得られたデー
タとxと一定値 α を加算した後、 2^{32} で割った余
りをとるという処理

を組み合わせることににより実現し、

暗号変換されるデータの転置変換処理を、

$$\text{Rot}_4(x),$$

第1図において、64ビットの平文101と
64ビット $\times 4 = 256$ ビットの鍵データ100
が32ビットマイクロコンピュータに入力され、
その後、プログラム103内の命令の順に32ビ
ットマイクロコンピュータ102において暗号変
換され、その結果として64ビットの暗号文10
4が出力される。

第2図は、第1図の32ビットマイクロコンピ
ュータ102とプログラム103において実行さ
れる暗号変換処理のフローを示している。

201: 入力されたデータMは上位32ビット
 M_1 と下位32ビット M_2 に分割される。

202: M_1 と M_2 のビット対応の排他的論和が
とられる。

$$\text{WORK 2} \leftarrow M_1 \oplus M_2$$

以下、+は同様の処理を示すものとする。

203: WORK 2と鍵データ K_1 のモジュロ
加算が行われる。

すなわち、32ビットのデータxを左または右へ
4ビット循環シフトするという処理と、

$$\text{Rot}_4(x),$$

すなわち、32ビットのデータxを左または右x
へ8ビット循環シフトするという処理と、

$$\text{Rot}_{16}(x),$$

すなわち、32ビットのデータxを左または右へ
16ビット循環シフトするという処理
を組み合わせることににより実現する。

(作用)

これにより、32ビットマイクロコンピュータ
を用いて、1回の基本命令で32ビットのデータ
が換字または転置されるので、暗号変換を高速に
行うことができる。

(実施例)

第1図は、本発明の一実施例である。

$$x \leftarrow \text{WORK 2} + K_1$$

ここに、 $x + K_1$ はxと K_1 の和を 2^{32} で割った余
りをとるという、 2^{32} を法としたモジュロ加算を
示している。

以下、+は同様の処理を示すものとする。

204: xを左へ2ビット循環シフトした後、
そのデータとxと1のモジュロ加算をとる。

$$x \leftarrow \text{Rot}_2(x) + x + 1$$

以下、 Rot_2 は同様の処理を示すものとする。

105: xを左へ4ビット循環シフトした後、
そのデータとxとの排他的論理和をとる。

$$x \leftarrow \text{Rot}_4(x) \oplus x$$

以下、 Rot_4 は同様の処理を示すものとする。

$$206: \text{WORK 1} \leftarrow x \oplus M_1$$

$$207: x \leftarrow x + K_2$$

208: $x \leftarrow \text{Rot}_2(x) + x + 1$

$y \leftarrow x$

209: $x \leftarrow \text{Rot}_8(x) \oplus x$

ここに、 $\text{Rot}_8(x)$ は x を左へ8ビット循環シフトさせることを示す。

210: $x \leftarrow x + K_3$

211: $x \leftarrow \text{Rot}_2(x) + x + 1$

212: $x \leftarrow \text{Rot}_{16}(x) + (x \wedge y)$

ここに、 $\text{Rot}_{16}(x)$ は x を左へ16ビット循環シフトすることを示す。また、 $x \wedge y$ は x と y とのビット対応の論理積をとることを示す。

213: $\text{WORK2} \leftarrow x \oplus \text{WORK2}$

214: $x \leftarrow \text{WORK2} + K_4$

215: $x \leftarrow \text{Rot}_2(x) + x$

216: $\text{WORK1} \leftarrow \text{WORK1} \oplus x$

217: $\text{WORK2} \leftarrow \text{WORK2} \oplus \text{WORK1}$

218: WORK1 を出力データの上位32ビット、 WORK2 を出力データの下位32ビットとして出力する。

以上、第2図に示すように関数 $\pi_1 \sim \pi_4$ を定義

$$C = \pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4 \cdot \pi_1 \\ \cdot \pi_4 \cdot \pi_3 \cdot \pi_2 \cdot \pi_1 (M)$$

としてもよい。

このとき、復号変換の式は

$$M = \pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4 \cdot \pi_1 \\ \pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4 \cdot \pi_1 (M)$$

である。

同様に、一掃に本実施例を n 回繰り返したものを暗号変換としてもよい。

実施例の変形例2

第4図は、本発明の他の実施例である。

401: 入力されたデータ M は上位16ビット M_1 と下位16ビット M_2 に分割される。

402: M_1 と M_2 のビット対応の排他的論理和がとられる。

すると、本実施例は、

$$C = \pi_1 \cdot \pi_4 \cdot \pi_3 \cdot \pi_2 \cdot \pi_1 (M)$$

というように合成関数で表すことができる。

関数 $\pi_i \cdot \pi_i$ ($i = 1 \sim 4$) は、すべて、

$$\pi_i \cdot \pi_i (x) = x$$

というように同じ関数変換を2回繰り返すともとに戻るという性質がある。

したがって、復号関数として、

$$M = \pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4 \cdot \pi_1 (C)$$

を用いれば、暗号文 C をもとの平文 M に戻すことができる。

実施例の変形例1

本実施例を2回繰り返したものを暗号変換として用いてもよい。すなわち、暗号変換を、

$$\text{WORK2} \leftarrow M_1 \oplus M_2$$

以下、 $+$ は同様の処理を示すものとする。

403: x と鍵データ K_1 のモジュロ減算が行われる。

$$x \leftarrow x - K_1$$

ここに、 $x - K_1$ は x と K_1 の差を 2^{16} で割った余りをとるという、 2^{16} を法としたモジュロ減算を示している。

以下、 $-$ は同様の処理を示すものとする。

404: x を左へ2ビット循環シフトした後、そのデータと 1 のモジュロ減算を行う。

$$x \leftarrow \text{Rot}(x) - x - 1$$

以下、 Rot_2 は同様の処理を示すものとする。

405: x を左へ4ビット循環シフトした後、そのデータと x との排他的論理和をとる。

$$x \leftarrow \text{Rot}_4(x) \oplus x$$

以下、 Rot_4 は同様の処理を示すものとする。

$$406: \text{WORK1} \leftarrow x \oplus M_1$$

$$407: x \leftarrow x \leftarrow K_2$$

$$y \leftarrow x$$

$$408: x \leftarrow \text{Rot}_2(x) - x - 1$$

$$409: x \leftarrow \text{Rot}_8(x) - (x \wedge y)$$

ここに、 $\text{Rot}_8(x)$ は x を左へ8ビット循環シフトすることを示す。また、 $x \wedge y$ は x と y とのビット対応の論理積をとることを示す。

$$410: \text{WORK2} \leftarrow x \oplus \text{WORK2}$$

$$411: x \leftarrow \text{WORK2} - K_2$$

$$412: x \leftarrow \text{Rot}_2(x) - x - 1$$

$$413: \text{WORK1} \leftarrow \text{WORK1} \oplus x$$

$$414: \text{WORK2} \leftarrow \text{WORK2} \oplus \text{WORK1}$$

415: WORK1を出力データの上位16ビット、WORK2を出力データの下部16ビットとして出力する。

504: x を左へ2ビット循環シフトした後、そのデータと x と1のモジュロ加算を行う。

$$x \leftarrow \text{Rot}_2(x) + x + 1$$

以下、 Rot_2 は同様の処理を示すものとする。

$$505: x \leftarrow \text{Rot}_4(x) + (x \wedge y)$$

ここに、 $\text{Rot}_4(x)$ は x を左へ4ビット循環シフトすることを示す。また、 $x \wedge y$ は x とのビット対応の論理積をとることを示す。

$$506: \text{WORK1} \leftarrow \text{WORK1} \oplus x$$

$$507: x \leftarrow \text{WORK1} + K_1$$

$$508: x \leftarrow \text{Rot}_4(x) + x + 1$$

$$509: \text{WORK2} \leftarrow \text{WORK2} \oplus x$$

$$510: \text{WORK1} \leftarrow \text{WORK1} \oplus \text{WORK2}$$

511: WORK1を出力データの上位8ビット、WORK2を出力データの下部8ビットとして出力する。

実施例の変形例4

第6図は本発明の他の一実施例である。

実施例の変形例3

第5図は、本発明の他の実施例である。

501: 入力されたデータ M は上位8ビット M_1 と下部8ビット M_2 に分割される。

502: M_1 と M_2 のビット対応の排他的論和がとられる。

$$\text{WORK2} \leftarrow M_1 \oplus M_2$$

以下、 $+$ は同様の処理を示すものとする。

503: x と鍵データ K_1 のモジュロ加算が行われる。

$$x \leftarrow \text{WORK2} + K_1$$

$$y \leftarrow x$$

ここに、 $x + K_1$ は x と K_1 の差を 2^8 で割った余りをとるという、 2^8 を法としたモジュロ加算を示している。

以下、 $+$ は同様の処理を示すものとする。

(1) 認証を行うメッセージ62を鍵データとして、任意の初期値61を本発明によるアルゴリズム63を用いて暗号化する。

(2) 暗号結果64を、(1)において用いたメッセージの続きのデータにより再び暗号化し、メッセージの終わりまでこの操作を繰り返す。

(3) 最終的な暗号結果をメッセージ認証コード65として出力する。

実施例の変形例5

第7図は本発明の他の実施例である。本ICカードは、第7項記載の方式によりメッセージの認証コードを生成する。

(1) メッセージの認証を行うために必要な初期値76をI/O74を通して、ICカード71内のマイクロコンピュータ72に送信する。

(2) 認証を行うメッセージ77を(1)と同様にマイクロコンピュータ72に順次送信し、マイクロコンピュータ72は、メモリ73に記憶されている暗号ソフト75により認証コード78を生成する。

(効果)

本実施例は、第3図に示すような換字、転置の繰返しを行っている。

つまり、第2図に示す実施例、

(203, 204), (207, 208),

(210, 211), (214, 215)の処理は、

$$x \leftarrow x + Ki$$

$$x \leftarrow \text{Rot}_2(x) + (x) + 1$$

の形となっており、これは、それぞれ、32ビットのデータを4ビットずつのブロックに分割したとき、各ブロック単位の換字処理を、上記2回のデータ変換により8ブロック分一斉に行っていると見ることができる。

ここに、4ビットのブロックデータ

$A = (a_1, a_2, a_3, a_4)$ 、ただし、

$a_i = 1 \text{ or } 0 \ (i = 1 \sim 4)$

が、

の処理を行っており、これらは、それぞれ、

(1) 4ビット左循環シフトを行うという転置を行った後、さらに換字を行うという処理、

(2) 8ビット左循環シフトを行うという転置を行った後、さらに換字を行うという処理、

(2) 16ビット左循環シフトを行うという処理を示している。

第3図から明らかなように、最初の32ビットのデータのうち、いかなるビットの変化も最後の32ビットのデータすべてに影響を与えることが分かる。

これにより、本実施例は、高度なランダム性を持つ暗号変換を効率良く行うという効果が得られることが分かる。

4. 図面の簡単な説明

第1図は、本発明を実施する暗号変換装置の一実施例、第2図は、第1図における暗号変換の詳細を示すフローチャート、第3図は、本発明の実施例が効率的に換字変換、転置変換を繰り返していることを示す説明図、第4図は、16ビットマ

$B = (b_1, b_2, b_3, b_4)$ 、ただし、

$b_i = 1 \text{ or } 0 \ (i = 1 \sim 4)$

に換字変換されるということは、

ブール代数の演算 f_1, f_2, f_3, f_4 が存在して、

$$b_1 = f_1(a_1, a_2, a_3, a_4)$$

$$b_2 = f_2(a_1, a_2, a_3, a_4)$$

$$b_3 = f_3(a_1, a_2, a_3, a_4)$$

$$b_4 = f_4(a_1, a_2, a_3, a_4)$$

となることを示す。

また、第2図の205、209、212はそれぞれ、

$$(1) \ x \leftarrow \text{Rot}_4(x) \oplus x$$

$$(2) \ x \leftarrow \text{Rot}_8(x) \oplus x$$

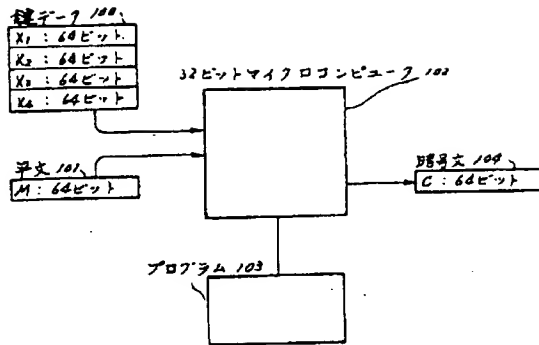
$$(3) \ x \leftarrow \text{Rot}_{16}(x) + (x \wedge y)$$

マイクロコンピュータを用いた場合の暗号変換の詳細を示すフローチャート、第5図は、8ビットマイクロコンピュータを用いた場合の暗号変換の詳細を示すフローチャート、第6図は、本発明による暗号アルゴリズムを用いてメッセージ認証コードを生成する方法を示すフローチャート、第7図は、本発明による暗号アルゴリズムを用いてメッセージ認証コードを生成するICカードの構成図である。

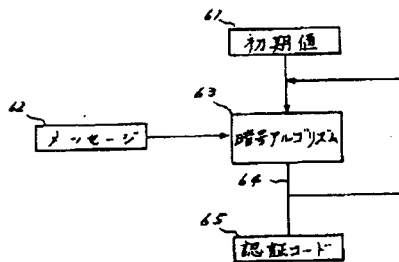
代理人 井理士 小川 勝 男



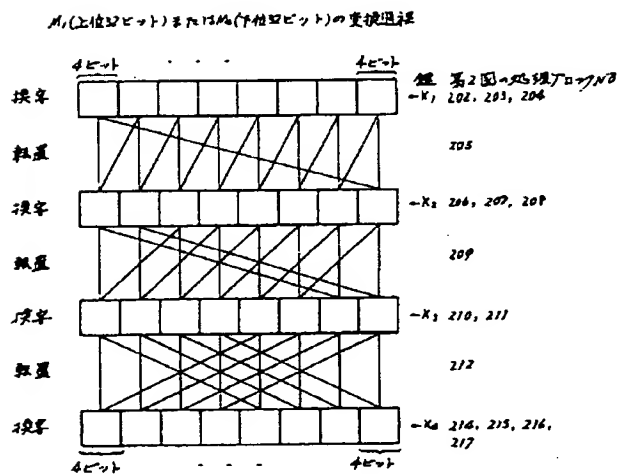
第 1 図



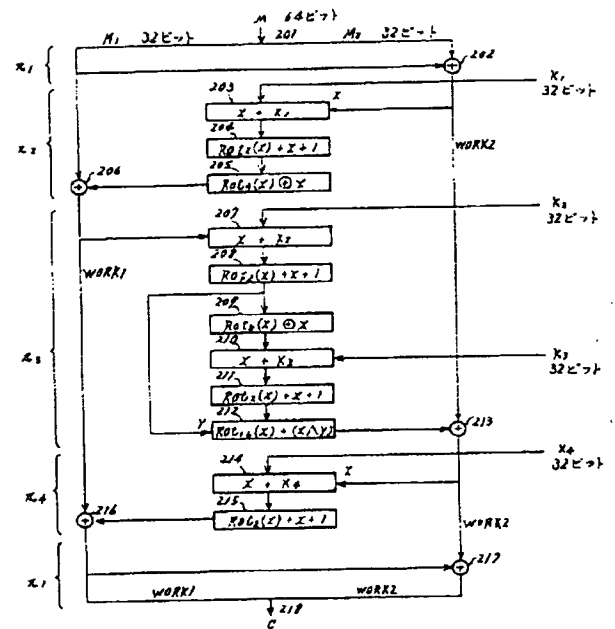
第 6 図



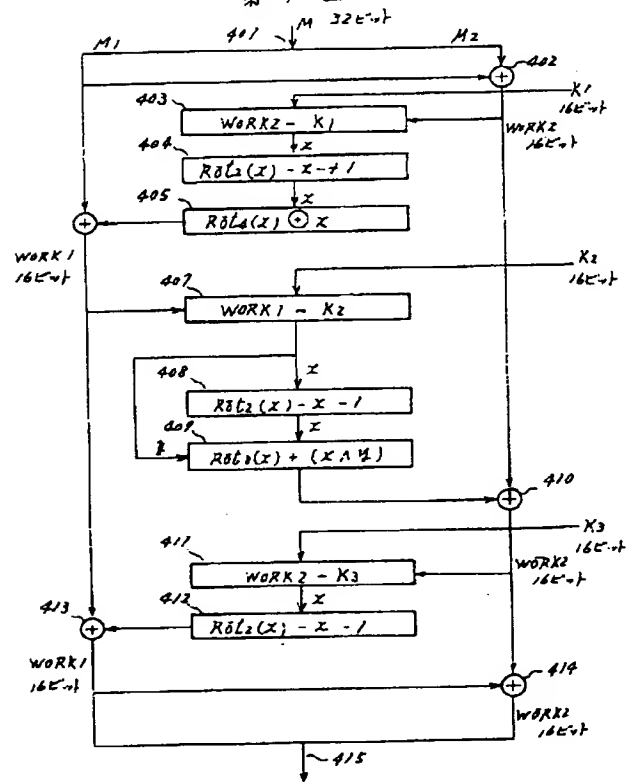
第 3 図



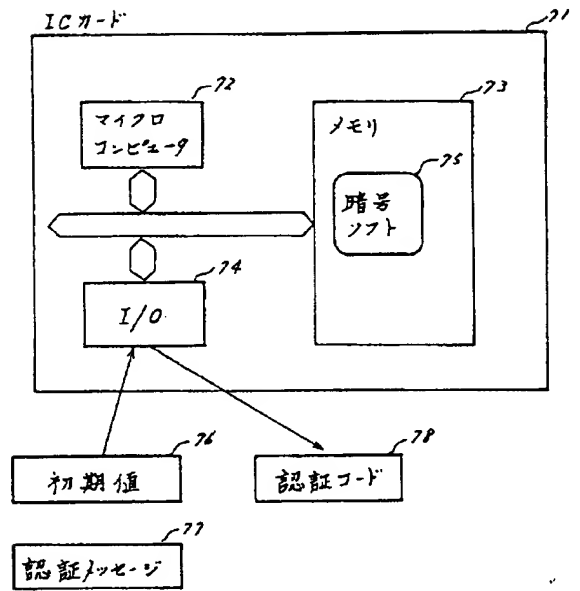
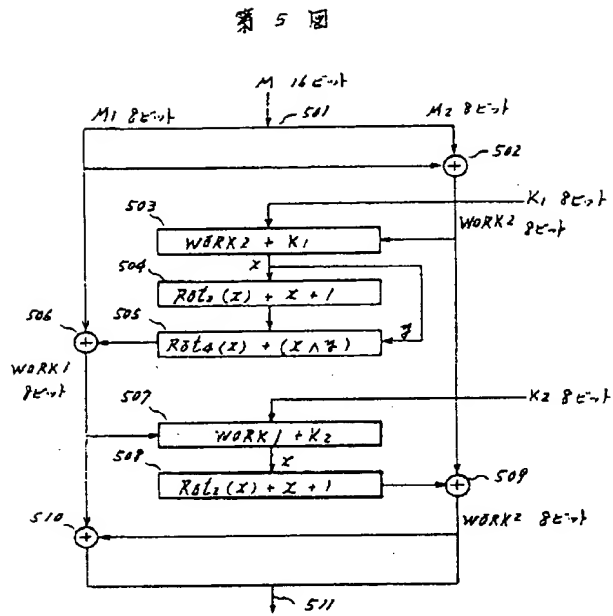
第 2 図



第 4 図



第 7 図



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第2区分

【発行日】平成8年(1996)7月12日

【公開番号】特開平1-276189

【公開日】平成1年(1989)11月6日

【年通号数】公開特許公報1-2762

【出願番号】特願昭63-103919

【国際特許分類第6版】

G09C 1/00

9364-5L

手 続 補 正 書

平成 7 年 4 月 28 日

特 許 庁 長 官 殿

事 件 の 展 示

昭和 63 年 特 許 願 第 103919 号

発 明 の 名 称 暗号化方法及び暗号の復号化方法

補 正 を す る 者

事件との関係 特 許 出 願 人
名 称 (510) 株式会社 日 立 製 作 所
(ほか1名)

代 理 人

居 所 〒100 東京都千代田区丸の内一丁目5番1号
株式会社 日 立 製 作 所 内
電 話 東 京 3212-1111(大代表)
氏 名 (6850) 丹 羽 士 小 川 勝 男

補 正 の 対 象 明細書の「発明の名称」の欄、「特許請求の範囲」の欄、
及び「発明の詳細な説明」の欄。

補正の内容

1. 明細書の発明の名称の欄の記載を「暗号化方法及び暗号の復号化方法」に訂正する。
2. 明細書の特許請求の範囲の欄の記載を別紙のとおりに補正する。
3. 明細書の発明の詳細な説明の欄について以下の補正を行う。
(1) 明細書第16頁第13行「本実施例を」を「上記実施例における変換関数 x_i から x_n までにあたる処理を」に訂正する。
(2) 明細書第17頁第1～2行の式を次のとおりに訂正する。
$$C = x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7 \cdot x_8 \cdot x_9 \cdot x_{10} \cdot x_{11} \cdot x_{12} \cdot x_{13} \cdot x_{14} \cdot x_{15} \cdot x_{16} \cdot x_{17} \cdot x_{18} \cdot x_{19} \cdot x_{20} \cdot x_{21} \cdot x_{22} \cdot x_{23} \cdot x_{24} \cdot x_{25} \cdot x_{26} \cdot x_{27} \cdot x_{28} \cdot x_{29} \cdot x_{30} \cdot x_{31} \cdot x_{32} \cdot x_{33} \cdot x_{34} \cdot x_{35} \cdot x_{36} \cdot x_{37} \cdot x_{38} \cdot x_{39} \cdot x_{40} \cdot x_{41} \cdot x_{42} \cdot x_{43} \cdot x_{44} \cdot x_{45} \cdot x_{46} \cdot x_{47} \cdot x_{48} \cdot x_{49} \cdot x_{50} \cdot x_{51} \cdot x_{52} \cdot x_{53} \cdot x_{54} \cdot x_{55} \cdot x_{56} \cdot x_{57} \cdot x_{58} \cdot x_{59} \cdot x_{60} \cdot x_{61} \cdot x_{62} \cdot x_{63} \cdot x_{64} \cdot x_{65} \cdot x_{66} \cdot x_{67} \cdot x_{68} \cdot x_{69} \cdot x_{70} \cdot x_{71} \cdot x_{72} \cdot x_{73} \cdot x_{74} \cdot x_{75} \cdot x_{76} \cdot x_{77} \cdot x_{78} \cdot x_{79} \cdot x_{80} \cdot x_{81} \cdot x_{82} \cdot x_{83} \cdot x_{84} \cdot x_{85} \cdot x_{86} \cdot x_{87} \cdot x_{88} \cdot x_{89} \cdot x_{90} \cdot x_{91} \cdot x_{92} \cdot x_{93} \cdot x_{94} \cdot x_{95} \cdot x_{96} \cdot x_{97} \cdot x_{98} \cdot x_{99} \cdot x_{100}$$

(3) 同頁第5～6行の式を次のとおりに訂正する。
$$M = x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7 \cdot x_8 \cdot x_9 \cdot x_{10} \cdot x_{11} \cdot x_{12} \cdot x_{13} \cdot x_{14} \cdot x_{15} \cdot x_{16} \cdot x_{17} \cdot x_{18} \cdot x_{19} \cdot x_{20} \cdot x_{21} \cdot x_{22} \cdot x_{23} \cdot x_{24} \cdot x_{25} \cdot x_{26} \cdot x_{27} \cdot x_{28} \cdot x_{29} \cdot x_{30} \cdot x_{31} \cdot x_{32} \cdot x_{33} \cdot x_{34} \cdot x_{35} \cdot x_{36} \cdot x_{37} \cdot x_{38} \cdot x_{39} \cdot x_{40} \cdot x_{41} \cdot x_{42} \cdot x_{43} \cdot x_{44} \cdot x_{45} \cdot x_{46} \cdot x_{47} \cdot x_{48} \cdot x_{49} \cdot x_{50} \cdot x_{51} \cdot x_{52} \cdot x_{53} \cdot x_{54} \cdot x_{55} \cdot x_{56} \cdot x_{57} \cdot x_{58} \cdot x_{59} \cdot x_{60} \cdot x_{61} \cdot x_{62} \cdot x_{63} \cdot x_{64} \cdot x_{65} \cdot x_{66} \cdot x_{67} \cdot x_{68} \cdot x_{69} \cdot x_{70} \cdot x_{71} \cdot x_{72} \cdot x_{73} \cdot x_{74} \cdot x_{75} \cdot x_{76} \cdot x_{77} \cdot x_{78} \cdot x_{79} \cdot x_{80} \cdot x_{81} \cdot x_{82} \cdot x_{83} \cdot x_{84} \cdot x_{85} \cdot x_{86} \cdot x_{87} \cdot x_{88} \cdot x_{89} \cdot x_{90} \cdot x_{91} \cdot x_{92} \cdot x_{93} \cdot x_{94} \cdot x_{95} \cdot x_{96} \cdot x_{97} \cdot x_{98} \cdot x_{99} \cdot x_{100}$$

以 上

別紙

特許請求の範囲

1. 暗号化対象データに対して所定ビット長のブロックごとに暗号化処理を行う暗号化方法において、該データをそれぞれ用いてデータへ換字処理を施すこと、およびデータ復乱のためデータに転置処理を施すことを交互に実施し、上記転置処理は、 2^n ビット ($n=2$ 以上の整数) だけ右もしくは左に処理中のデータを循環シフトする操作を含むことを特徴とする暗号化方法。
2. 上記転置処理における循環シフトの深さは換字処理と転置処理の交互実施ごとに 2^1 ビット、 2^2 ビット、 \dots 、と順次増大することを特徴とする特許請求の範囲第1項記載の暗号化方法。
3. 上記転置処理は、3 2 ビットのデータを4 ビット、または8 ビット、または16 ビットだけ右もしくは左に循環シフトする操作を含むことを特徴とする特許請求の範囲第1項記載の暗号化方法。
4. 上記換字処理と転置処理を組み合わせた暗号化処理は、暗号化されたデータと、次の暗号化されるべきデータとの合いだて演算処理を施した後、該演算処理結果をさらに暗号化するというフィードバック処理を有することを特徴とする特許請求の範囲第1項記載の暗号化方法。
5. 該データをそれぞれ用いてデータへ換字処理を施すこと、およびデータ復乱のためデータに転置処理を施すことを交互に実施し、上記転置処理は、 2^n ビット ($n=2$ 以上の整数) だけ右もしくは左に処理中のデータを循環シフトする操作を含む所定の暗号化方法を用い、デジタル署名により認証すべきメッセージを順次切り出して上記換データとし、所定の初期値を上記所定の暗号化方法で順次暗号化してメッセージ承認コードとするメッセージの認証方法。
6. 該データをそれぞれ用いてデータへ換字処理を施すこと、およびデータ復乱のためデータに転置処理を施すことを交互に実施し、上記転置処理は、 2^n ビット ($n=2$ 以上の整数) だけ右もしくは左に処理中のデータを循環シフトする操作を含む所定の暗号化アルゴリズムをICカードの記憶領域に記憶し、上記ICカードでは認証すべきメッセージと初期データとが入力すると、該メッセージを順次切り出して上記換データとし、上記初期データに対し上記所定の

暗号化アルゴリズムによる暗号化処理を順次施してメッセージ承認コードを発生することを特徴とするメッセージの認証方法。

7. 該データをそれぞれ用いてデータへ換字処理を施すこと、およびデータ復乱のためデータに転置処理を施すことを交互に実施し、上記転置処理は、 2^n ビット ($n=2$ 以上の整数) だけ右もしくは左に処理中のデータを循環シフトする操作を含む所定の暗号化方法で作成された暗号を復号する復号化方法であり、上記所定の暗号化方法の処理ステップ全体をそれぞれ同一の関数変換を2度行えばデータが元に戻る関数変換の逆数変換の繰り返しとしたとき、上記作成された暗号に対し、上記逆数変換の関数変換を上記暗号化方法とは逆の順序で実行することを特徴とする暗号の復号化方法。